Constraint-based Routing for Ad-hoc Networks

Yi Shang
Dept. of Computer Engineering/Science
Univ. of Missouri-Columbia
Columbia, MO 65211
shangy@missouri.edu

Markus P.J. Fromherz, Ying Zhang and Lara S. Crawford
Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304, U.S.A.
{fromherz,yzhang,lcrawford}@parc.com

***Abstract*— Future large-scale networks, such as sensor networks, will consist of hundreds and even thousands of wirelessly connected sensor and actuator nodes. The nodes are typically highly resource-constrained (processor, memory, and power), have limited communication range, and are prone to failure. Furthermore, there is no global information about the exact location and connectivity of the nodes. Consequently, the explicit consideration of network and task constraints and objectives will be an important part of routing algorithms for these networks. In this paper, we present a distributed constraint-based routing approach that represents destination conditions as well as routing constraints and objectives explicitly. We further present an efficient routing algorithm, CB-LRTA*, that extends traditional Learning Real-Time A* (LRTA*) with back-propagation. We evaluate CB-LRTA* using simulation and demonstrate that it improves convergence to the optimal route over LRTA*.**

I. INTRODUCTION

Large-scale networks with hundreds and even thousands of very small, battery-powered and wirelessly connected sensor and actuator nodes are becoming a reality [4]. The sheer number of devices in such networks, the resource constraints of the nodes, and the dynamics of the environment call for adaptive, robust, scalable, and constraint-aware algorithms for discovery, routing, and aggregation of information. The algorithms should be localized, use minimal storage, adapt to changes, and have minimal communication cost.

Imagine a network of sensors sprinkled across a large building or an area such as a forest. Typical tasks considered for such networks are to send a message to a node at a given location (without knowing which node or nodes are there, or how to get there), to retrieve sensor data (e.g., sound or temperature levels) from nodes in a given region, and to find nodes with sensor data in a given range without any geographical information. In some cases, one is interested in passing on a single message and thus simply wants to find a good path, maybe sub-optimal, to the destination quickly, since finding the optimal path could be much more time and energy consuming. In other cases, many packets are to be sent between source and destination (e.g., updates on sensor readings), and it would be better to establish a lowest cost path or the shortest path between them. When the energy levels of the nodes are very limited, multiple different paths may be preferable to distribute energy usage and prolong the lifetime of the network. Some applications add further requirements to message routing, such as avoiding compromised regions of the network in military applications.

In the past, algorithms proposed for discovery and routing tasks in sensor networks [7], [12], especially those with energy awareness, use a variant of dynamic programming, in particular search algorithms related to Learning Real-Time A* (LRTA*) [9], [10].

Implicitly, these algorithms encode various constraints and objectives about the task, often in a task-specific manner. Task representations and routing algorithms are inevitably linked and can be hard to change as the requirements of the task change. In this paper, we propose a constraint-based approach for routing in ad-hoc networks. It consists of a generic, constraint-based task representation of the discovery problem and constraint-aware routing protocols. The approach is a generalization of existing sensor network routing and geographical routing protocols. The representation does not assume a particular routing algorithm, and in fact the choice of algorithm can be embedded in the message as well. In addition, we present a variant of LRTA*, Constraint-based Back-propagating LRTA* (CB-LRTA*), which extends traditional LRTA* in three ways: explicit inclusion of hard constraints, node heuristics in addition to connection heuristics, and parallel learning with a distributed memory.

II. RELATED WORK

A. Ad-hoc routing protocols

Many previous ad-hoc routing protocols require greater energy resources of the nodes and higher bandwidth than what is available in sensor networks. For example, Dynamic Source Routing (DSR) [6] floods a route request packet throughout the network. Location Aided Routing (LAR) [8] improves DSR and uses geographic location information to limit the route request flooding to a smaller region, where it is most probable the destination is located.

Geographic location information has been used to develop efficient, scalable routing protocols. Geographic routing allows routers to be stateless and requires propagation of topology information for only a single hop. Most geographic ad-hoc routing protocols use greedy algorithms to forward the packet to the destination. They differ in how they recover when greedy forwarding is impossible, e.g., at communication holes or when avoiding obstacles.

Finn [2] proposes flooding search for recovering from local maxima. Karp and Kung propose Greedy Perimeter Stateless

0-7803-7724-9/03/$17.00

Routing (GPSR) [7] to have better scalability. GPSR recovers from local maxima by deriving a planar graph out of the original network graph and then routing around the perimeter of the region containing a local maximum. A drawback of GPSR is that it tends to concentrate traffic on the perimeter when it routes around holes or obstacles, thus burning out the nodes on the perimeter sooner.

Geographical and Energy Aware Routing (GEAR) [12] achieves good energy efficiency. GEAR is based on a real-time heuristic search method, Learning Real-Time A* (LRTA*) [10]. It uses energy aware and geographically informed neighbor selection to route a packet towards the target region. The strategy attempts to balance energy consumption and thereby increase network lifetime. In related work, CADR [1] uses a sophisticated information metric derived from sensor data to guide the routing process.

Directed diffusion [5] is a data-centric paradigm for sensor network applications. All communication is for named data and all nodes are application aware. This enables diffusion to achieve energy savings by selecting good paths. It uses initial and periodic data flooding throughout the network. Data generated by sensor nodes is named using attribute-value pairs.

An issue common to these algorithms is that expectations about both the task (e.g., find the destination) and algorithm properties (e.g., conserve energy) are built into the algorithms and cannot be changed easily. Since routing algorithms cannot be uploaded repeatedly onto thousands of already deployed nodes, a more general, programmable approach is highly desirable.

B. *Real-time search methods*

Real-time (heuristic) search methods interleave planning and plan execution and restrict planning to a local area. LRTA* is a popular real-time search method [11]. It not only acts in real time, but also converges to a shortest path when it solves the same planning task repeatedly. A related algorithm is FALCONS [3], which differs in the selection of successors and has been shown to converge to a shortest path faster than LRTA*.

Starting from a source node, LRTA* traverses the graph in search of the destination by using local information only. Concretely, at current node i, a decision is made about which node to move on to next based on information about the node and its neighbors (Fig. 1). This information includes the known cost $c(i, j)$ to move to node j and the estimated cost $h(j)$ to move from node j to the destination. (Costs are usually distances.) Node i chooses neighbor j with minimum value $f(j) = c(i, j) + h(j)$, updates its own heuristic value $h(i)$ to $f(j)$, and moves on to node j.

Real-time search methods similar to LRTA* differ in two dimensions: the size of their local search spaces and the informedness of the initial state values. For exploration and goal-directed search in unknown terrain, the local search space may contain only the current state or all the known (visited) parts of the state space. Heuristic information may or may not be available to estimate the state values for goal-directed search.

```
received ($s$, find($goal$)) at node $i$ do
    if found($goal$) then return; end
    for all $j \in N_i$ do $f(j) \leftarrow c(i, j) + h(j)$; end
    $j \leftarrow \text{argmin}_{j \in N_i} f(j)$;
    $h(i) \leftarrow f(j)$;
    send($j$, find($goal$));
end
```

Fig. 1. LRTA* – Nodes i receive find request messages, compute successors j among their neighbors N_i, and forward the messages to these successors. The command send(j, m) in node i sends message m to node j, which leads to event received(i, m) in node j. In a distributed network, multiple nodes may execute this code in parallel.

III. CONSTRAINT-BASED ROUTING

We make the following assumptions about a distributed sensor and actuator network: The system consists of a large number of nodes that are distributed geographically, each able to communicate with a small subset of neighbor nodes, and equipped with a processor and a number of sensors (e.g., light, sound, temperature). Nodes know their own items of interests and locations, as well as those of their neighbors (e.g., through a publish-and-subscribe connection), but they have no knowledge about nodes outside their neighborhood. Items of interest include sensor values, battery level, available memory, etc.

Typical tasks are sending and retrieving data between nodes. There are constraints both on the destinations and the routes: finding nodes in a certain geographic region or with sensor readings in a given range or avoiding nodes with low battery levels or near loud objects. The goal is to find the shortest route to a destination node while also satisfying the route constraints and optimizing the route objectives.

A. *Constraint-based Task Representation*

A sensor and actuator network is defined by a graph $\langle V, E \rangle$, with V the set of n nodes and E the set of edges. Each node i has a set of *neighbors* $N_i \subseteq V$ with which it can communicate.

Nodes are described by sets of *attributes*. These may denote any interesting node characteristic, such as node position and sensor values. Data is generated by sensor nodes in the form of attribute-value pairs. For example, a node i has positions x_i and y_i, light sensor value l_i, temperature sensor value t_i, energy level e_i, etc.

Specifically, a routing task $T(V, E, s, O_d, C_r, O_r)$ is defined by a network graph $\langle V, E \rangle$, a source node s, a set of destination objectives O_d, a set of route constraints C_r, and a set of route objectives O_r. Starting at the source node s, the goal of the search is to find a destination node that minimizes the destination objectives, as well as a route that satisfies the route constraints and minimize the route objectives. Constraints and objectives are defined on node attributes.

Examples for destination objectives are *location-based* objectives, e.g., finding the node in region R, and *sensor-based* objectives, e.g., finding a node with temperature level below t_d.

Examples for route constraints are constraints to avoid obstacles or “risky” areas, e.g., avoid nodes in the light, $C_r = \{l \leq l_t\}$, where l_t is a threshold below which the node is considered to be in the dark, or constraints to use only nodes with energy level above a certain level e_t, $C_r = \{e \geq e_t\}$. Route objectives can be used to describe preferences between alternative paths, e.g., prefer nodes with high remaining energy levels, $O_r = \{e_{\max} - e\}$. where $e_{\max}$ is the maximum possible energy level.

The requesting node initiates an interest in the form of $T(V, E, s, O_d, C_r, O_r)$ and sends it to one of its neighbor nodes. Intermediate nodes in the network forward the message until it reaches a destination. Each node maintains an interest cache, containing one entry for each distinct interest. Each entry has several fields: destination objectives, routing constraints, routing objectives, time stamp of the last received matching interest, parent field specifying where the interest comes from and requested data rate, a duration field indicating the approximate lifetime of the interest, information about its neighbor nodes, such as heuristic values estimating the best cost going through a neighbor node, and others.

*B. Heuristic Functions in LRTA**

Route discovery is carried out by agent-based search algorithms, such as LRTA*. The heuristic function used for selecting successor nodes is $f(j) = c(i, j) + h(j)$ where the heuristic function $h(j)$ estimates the true cost from node j to a destination node.

Let $c(i, j)$ represent the total cost of going from node i to j, which consists of the route objective values of node j and the costs of moving a message to node j from node i and processing the message at node j. Assuming the message transmission and processing costs are constant C, i.e., the same for all nodes, we have

$$c(i, j) = C + \sum_{o_l \in O_r} w_l o_l(j),$$

where w_l are weights.

There are various h functions. When the cost of future nodes cannot be estimated, we set $h(j)$ to 0. Alternatively, we can use the route objectives in h, such as in a weighted sum form

$$h(j) = \sum_{o_k \in O_d} w_k o_k(j) \tag{1}$$

where $o_k(j)$ is the value of an destination objective o_k at node j and w_k is a weight. Examples of objective functions are $|x(j) - p|$ (the coordinate of the destination is at p), and $\max(x(j) - u, 0) + \max(l - x(j), 0)$ (the coordinate of the destination is between l and u). The objectives may be normalized by the attribute domain ranges.

As an example, consider the following task:

$$O_d = \{|x - x_d|, |y - y_d|\}, C_r = \{l \leq l_t\}, O_r = \{e_{\max} - e\}. \tag{2}$$

```
start ($id, O_d, C_r, O_r$) at node $i$ do
    record source($id$);
    send($i$,find($id, O_d, C_r, O_r$));
end

received ($s$,find($id, O_d, C_r, O_r$)) at node $i$ do
    if no record sender($id$) exists then
        record (sender($id$) = $s$);
    end
    if the values of $O_d$ at $i$ satisfy stopping criteria then
        send(sender($id$),back($id, O_d, C_r, O_r$));
        return;
    end
    $N'_i \leftarrow \{j \in N_i \mid$ satisfied$(C_r, j)\}$;
    for all $j \in N'_i$ do
        $f(j) \leftarrow c(i, j) + h(j)$;
    end
    $j \leftarrow$ argmin$_{j \in N'_i} f(j)$;
    $h(i) \leftarrow f(j)$;
    send($j$, find($id, O_d, C_r, O_r$));
end

received ($s$,back($id, O_d, C_r, O_r$)) at node $i$ do
    if source($id$) then return; end
    $N'_i \leftarrow \{j \in N_i \mid$ satisfied$(C_r, j)\}$;
    for all $j \in N'_i$ do
        $f(j) \leftarrow c(i, j) + h(j)$;
    end
    $j \leftarrow$ argmin$_{j \in N'_i} f(j)$;
    $h(i) \leftarrow f(j)$;
    send(sender($id$),back($id, O_d, C_r, O_r$));
end
```

Fig. 2. CB-LRTA* – Command start() in the source node starts the search. Command send(j, m) in node i leads to event received(i, m) in node j. Here, the message includes a message identifier id as well as the constraints and objectives. Since different messages with unrelated constraints and objectives may be passing through, the identifier is used to distinguish h entries for these different messages.

The initial value of $h(j)$ is $|x_j - x_d| + |y_j - y_d|$ and $f(j) = C + (e_{\max} - e_j) + h(j)$. The successor nodes must satisfy the route constraints in C_r, which are not included in f.

IV. CB-LRTA*

CB-LRTA* is a variation of LRTA* using the generalized, attribute-based heuristic functions for neighbor evaluation (Fig. 2). CB-LRTA* uses the route objectives and constraints in node selection and the destination objectives to determine termination. Decisions for forwarding a message are made locally at each node based on the task specification, O_d, C_r, and O_r.

Here, we illustrate the behavior of CB-LRTA* using some simple examples, assuming the nodes know their locations.

Example 1. $O_d = \{|x - x_d|, |y - y_d|\}, C_r = \{\}, O_r = \{\}$. CB-LRTA* becomes geographic routing and moves to a neighbor node closest to the destination.

Example 2. $O_d = \{x - x_d|, |y - y_d|\}, C_r = \{\}, O_r = \{e_{\max} - e\}$. CB-LRTA* tends to move to a neighbor node closer to the destination and with higher energy level. Its behavior is similar to GEAR.

Example 3. $O_d = \{\max(t - t_d, 0)\}, C_r = \{l \le l_t\}, O_r = \{e_{\max} - e\}$. CB-LRTA* tends to move to a neighbor node with lower temperature and higher energy level, which must have a light sensor value less than l_t.

Furthermore, CB-LRTA* includes a back-propagation mechanism that goes back from the destination to the source node along the forwarding route with redundant loops removed, updating heuristic values of the nodes on the way. As will be shown, this can significantly speed up convergence to the optimal route from source to destination. Back-propagation is particularly appropriate for tasks in sensor networks, where acknowledgments and retrieved sensor data often have to be sent back from destination to source anyway. By piggybacking the heuristic values on reverse messages, no additional packets need to be sent, and the amount of additional communication and computation can be kept minimal.

V. EXPERIMENTS

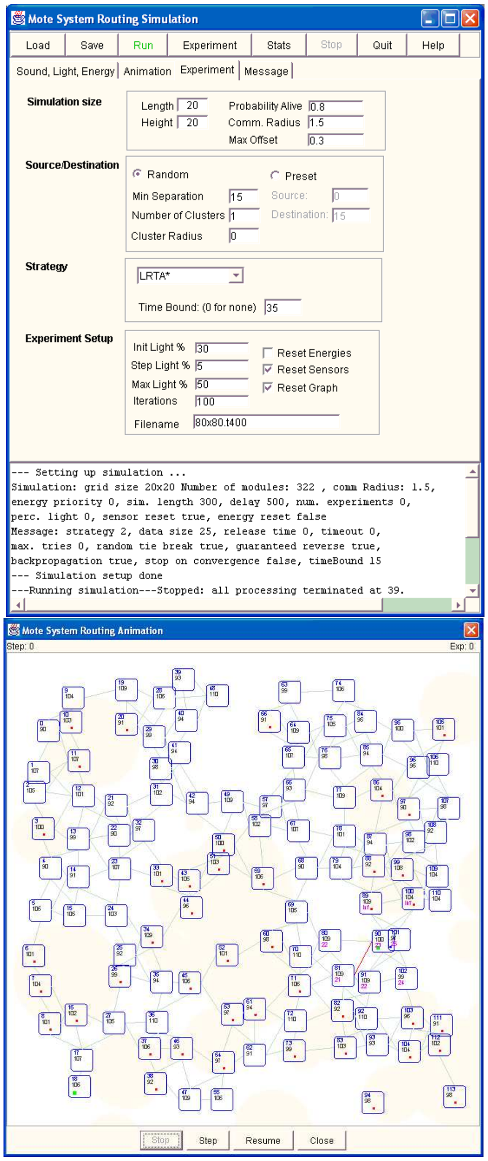
We call a "message" the data to be sent from source to destination, with acknowledgment sent back to the source. We call a "packet" the exchange between two nodes to forward a message. We count both the number of packets from source to destination ("forward") and the total number of packets for both forwarding and acknowledging a message ("total").

The performance measure is the cost incurred until convergence to the optimal route. All data is averaged over 100 runs per experiment, using a packet-level simulator shown in Fig. 3.

A. *Scaling with obstacles*

Fig. 4 shows the scaling of sending messages in a 400 node network with increasing number of obstacles. The task is defined as in Eq. (2) but without the route objectives. Obstacles (nodes with high light sensor values) are chosen randomly such that a path from source to destination exists.

Fig. 4a compares CB-LRTA* with LRTA* on the number of forward and total packets until convergence to the optimal path. The number of packets for the first path is shown for reference. The result shows that back-propagation significantly improves convergence, whether comparing forward or total number of packets. The total number of packets of CB-LRTA* is comparable to the forward number of packets of LRTA*. This makes CB-LRTA* competitive even in a non-distributed context, where LRTA* would restart directly at the source for multiple trials, while CB-LRTA* needs to trace the path back to the source. The poor performance of LRTA* in Fig. 4a is due to LRTA* needing many extra messages (up to twice as many) for small improvements before it converges. Finally, we observe at least one phase transition around 40%, from where it becomes easier to find the route due to the decreasing number of route options.



Simulation size
Length 20
Height 20
Probability Alive 0.8
Comm. Radius 1.5
Max Offset 0.3

Source/Destination
● Random
○ Preset
Min Separation 15
Number of Clusters 1
Cluster Radius 0
Source: 0
Destination: 15

Strategy
LRTA*
Time Bound: (0 for none) 35

Experiment Setup
Init Light % 30
Step Light % 5
Max Light % 50
Iterations 100
Filename 80x80.t400
☐ Reset Energies
☑ Reset Sensors
☑ Reset Graph

```
--- Setting up simulation ...
Simulation: grid size 20x20 Number of modules: 322 , comm Radius: 1.5,
energy priority 0, sim. length 300, delay 500, num. experiments 0,
perc. light 0, sensor reset true, energy reset false
Message: strategy 2, data size 25, release time 0, timeout 0,
max. tries 0, random tie break true, guaranteed reverse true,
backpropagation true, stop on convergence false, timeBound 15
--- Simulation setup done
---Running simulation---Stopped: all processing terminated at 39.
```

Fig. 3. Front end (upper) and animation (lower) of the routing simulation environment

B. *Scaling with size*

Fig. 5 shows the scaling of sending messages in networks with increasing size, while holding the ratio of obstacles constant at 30%. The same task as before is used. Again, CB-LRTA* and LRTA* are compared, showing the number of forward and total packets until convergence to the optimal path, as well as the number of packets for the first path.

Again, CB-LRTA* performs significantly better and the large difference between CB-LRTA* and LRTA* is primarily due to additional messages (or retries) needed by LRTA* before search converges.

C. *Performance with message interleaving*

A distributed network offers the option to release multiple messages with short delays instead of waiting for the

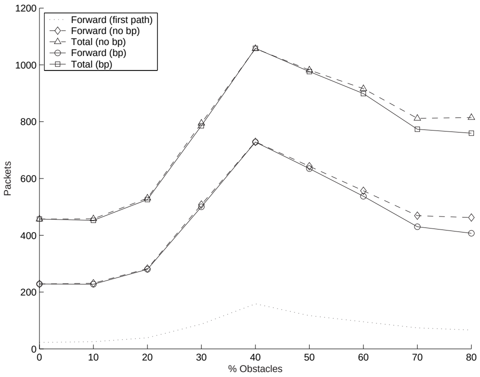


Fig. 4. Scaling with increasing number of obstacles: average number of packets until convergence with and without back-propagation (“bp”) (400 nodes, random source and destination nodes at least 20 hops apart)

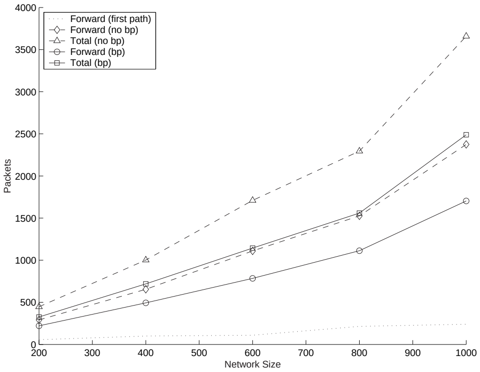


Fig. 5. Scaling with increasing number of nodes: average number of packets until convergence (random source and destination nodes at least $(w + l)/4$ hops apart (width w, length l), 30% random obstacles)

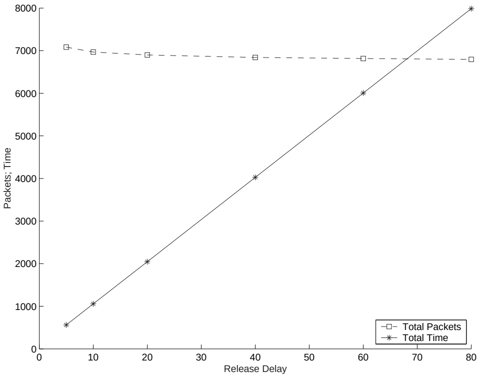


Fig. 6. Scaling with increasing release delay between messages: average total number of packets and simulation time for 100 messages and 30% random obstacles.)

acknowledgment of each message. Fig. 6 compares results with increasing release delay (in simulation time units) for 100 messages. The case of waiting for acknowledgment is not shown, but would be similar to a release delay of 80. As can be seen, shorter release delays incur a very small amount of overhead in the total number of packets to be sent, but the total time to deliver all the messages is significantly shorter.

D. Load-balancing through route objectives

Route objectives can effectively balance the usage of nodes in message routing. For example, without a route objective of using higher energy nodes, repeated messages are routed through the most direct path and quickly use up the nodes’ energies, which erects a significant barrier for messages between other source and destination nodes in the network. Adding the route objective, in contrast, leads to a much more preferable distribution of energy usage in the network.

VI. CONCLUSIONS

The research presented here is a first step towards a generic approach of representing and solving routing tasks in resource-constrained, ad-hoc networks as constraint problems. We present an explicit representation of routing and destination constraints and objectives, as well as CB-LRTA*, a constraint-based, real-time search method that extends LRTA* and converges faster than LRTA*. We have successfully implemented this method on a real sensor network.

In future work, an extension is to make the routing algorithm time-aware, so that secondary objectives (such as load-balancing) can be traded off against primary objectives (such as reaching the destination) when routing messages with deadlines. Furthermore, other real-time search algorithms, such as FALCONS, may be extended for routing applications.

VII. ACKNOWLEDGEMENTS

This work was partially supported by DARPA under contract F33615-01-C-1904.

REFERENCES

[1] M. Chu, H. Haussecker, and F. Zhao. Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. *Int. Journal on High Performance Computing Applications*, June 2002.

[2] G. G. Finn. Routing and addressing problems in large metropolitan-scale internetworks. Technical report isi/rr-87-180, USC/ISI, March 1987.

[3] D. Furcy and S. Koenig. Speeding up the convergence of real-time search. In *Proc. 17th National Conf. on Artificial Intelligence*, pages 891–897, 2000.

[4] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker. An empirical study of epidemic algorithms in large scale multihop wireless networks. Technical report ucla/csd-tr-02-0013, UCLA Computer Science Department, 2002.

[5] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proc. 6th Int’l Conf. on Mobile Computing and Networks (ACM Mobicom)*, Boston, MA, 2000.

[6] D. B. Johnson and D. B. Maltz. Dynamic source routing in ad hoc wireless networks. In T. Imielinski and H. Korth, editors, *Mobile Computing*, pages 153–181. Kluwer Academic, 1996.

[7] B. Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proc. 6th Int’l Conf. on Mobile Computing and Networks (ACM Mobicom)*, Boston, MA, 2000.

[8] Y. Ko and N. Vaidya. Location-aided routing in mobile adhoc networks. In *Proc. 4th Annual ACM/IEEE Int’l Conf. on Mobile Computing and Networks (Mobicom)*, Dallas, TX, 1998.

[9] S. Koenig. Agent-centered search. *AI Magazine*, 22(4):109–131, 2001.

[10] R. Korf. Real-time heuristic search. *Artificial Intelligence*, 42(2-3):189–211, 1990.

[11] M. Yokoo and T. Ishida. Search algorithms for agents. In G. Weiss, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, chapter 4, pages 165–199. The MIT Press, Cambridge, MA, 1999.

[12] Y. Yu, R. Govindan, and D. Estrin. Geographical and energy aware routing: a recursive data dissemination protocol for wireless sensor networks. Technical report ucla/csd-tr-01-0023, UCLA Computer Science Department, May 2001.